Konsolidierung einer Data Mart Farm auf eine ExaData X2-2

Oliver Scheibe, Jens Albrecht, Marc Fiedler
GfK Retail and Technology GmbH
Nordwestring 101, 90419 Nürnberg

Schlüsselworte:
ExaData X2-2, Data Marts, Auto DOP, Konsolidierung, OLTP-Kompression

Einleitung

Die GfK-Gruppe ist eines der weltweit führenden Marktforschungsunternehmen mit einem Gesamtumsatz von ca. 1,3 Milliarden € im abgelaufenen Geschäftsjahr 2010. Im Geschäftsfeld „Retail and Technology“ stellt die GfK ihren Kunden umfassende Marktberichte zu technischen Gebrauchsgütern auf internationaler Basis zur Verfügung. Der Kundenkreis der GfK Retail and Technology GmbH umfasst vornehmlich international operierende Markenartikelhersteller und Handelshäuser. Die Grundlage für das Reporting bilden Verkaufs-, Preis- und Bestandsdaten aus etwa 350.000 Geschäften in über 100 Ländern. Mehrere hundert Kunden erhalten nationale und internationale Berichte zu über 400 Warengruppen, die im interaktiven Betrieb online abgerufen werden können.
Durch den direkten Zugriff der Kunden auf die Daten ergeben sich sehr hohe Performance-Anforderungen, die auf einem zentralen System aufgrund paralleler Schreiblast bisher nicht erfüllt werden konnten. Aus diesem Grunde wurden Data Marts auf kostengünstiger Hardware eingerichtet, die konsequent auf eine optimale Nutzung des Hauptspeichers unter Oracle ausgerichtet sind. Diese Lösung stößt bei massivem parallelen Lesezugriff auf den Data Marts an ihre Grenze. Daher wird die Möglichkeit einer Konsolidierung der Data Mart Farm auf eine ExaData X 2-2 angestrebt.

StarTrack – Die Data Warehouse Plattform der GfK Retail and Technology

Die Verarbeitung der erhobenen Handelsdaten erfolgt durch das in Eigenentwicklung entstandene StarTrack System (*S*ystem *t*o *A*nalyze and *R*eport on *Track*ing Data). Der für das Reporting relevante Teil besteht aus einem zentralen Data Warehouse, der Reporting Base (Abb. 1), das Mitte 2011 von einem 6 Knoten RAC ebenfalls auf eine ExaData X2-2 umgestellt wurde.

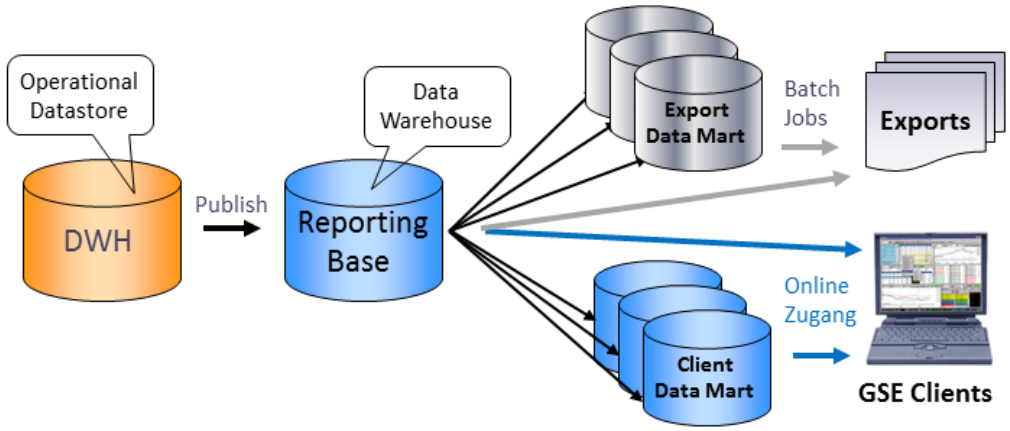

Abb. 1: StarTrack Reporting Infrastruktur

Die im Handel erhobenen Daten werden durch einen umfangreichen ETL-Prozess zu berichtsfertigen Daten für den Kundenzugriff auf der Reporting Base veröffentlicht. Dieser Prozessschritt wird als *Publish* bezeichnet. Für den *Publish* sind keine dedizierten Ladezeitpunkte vorgesehen, d.h. die Reporting Base und damit alle angeschlossenen Systeme werden im 24*7-Stunden-Betrieb geladen. Neben der reinen Veröffentlichung der neuesten Daten finden Korrekturen (*Republishes*) bereits veröffentlichter Daten, die großvolumige DML-Operationen auf der Reporting Base nach sich ziehen, statt. Zusätzlich laufen auf dem zentralen Data Warehouse hochvoluminöse Entladeprozesse, sog. Exports. Durch die permanenten Lade- und Entladevorgänge ist eine stabile Anfrageperformance auf der zentralen Reporting Base nicht möglich. Die Kunden, deren Erwartungen an die Ausführungszeiten für das interaktive Navigieren mit dem StarTrack Explorer (GSE) bei maximal zehn Sekunden liegen, werden daher auf Data Marts geleitet.
Bereits über 90 Prozent aller Reportausführungen liegen innerhalb der zehn Sekunden-Grenze, weitere fünf Prozent liegen unter 20 Sekunden und nochmals fünf Prozent benötigen mehr als 20 Sekunden. Erreicht werden diese Reportausführungszeiten mit einer Data Mart Farm bestehend aus 64 Bladeservern die jeweils einen Ausschnitt der Daten – meist zusammengehörige Warengruppen eines bestimmten *Sektors*, z.B. Consumer Electronics oder IT - des Zentralsystems enthalten. Über die konsequente Nutzung des Hauptspeichers, mit SGAs bis zu 72 GB und weiteren Maßnahmen, wird zeitintensiver I/O auf den Data Marts während des Reportings weitestgehend vermieden. Mit der Realisierung eines Schattenkonzepts, es existieren jeweils zwei Server pro Data Mart, können Schreib- und Lesevorgänge entkoppelt werden. Während auf einem Server das Reporting stattfindet, können auf dem Schattenserver neue bzw. veränderte Daten geladen werden (Abb. 2, Details in [2]).

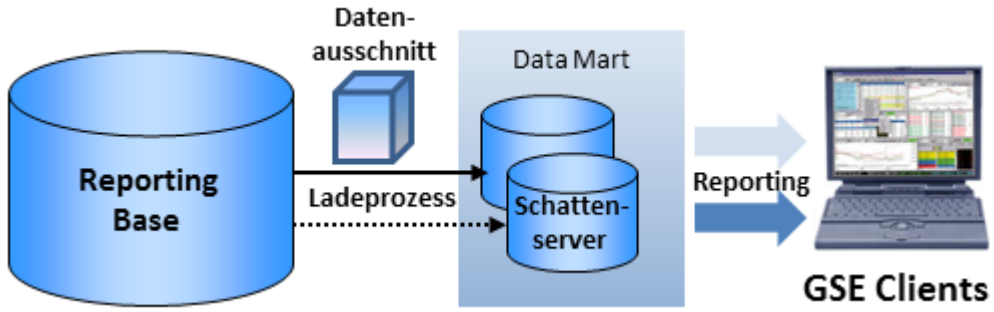

Abb. 2: StarTrack Reporting Infrastruktur

Allerdings skalieren die Data Marts bei gleichzeitigen Kundenzugriffen in der Leseperformance schlecht, da die Blade Server nur über ein sehr schwaches I/O-System verfügen. Die Kundenzugriffe müssen immer wieder auf neue Data Marts verteilt werden, was zu einem stetigen Wachstum der Data Mart Farm führt. Bei einer insgesamt sinkenden durchschnittlichen Auslastung der Data Mart Server.

Stetig wachsende Datenvolumina, steigende Kundenzugriffe, die Ausführung von immer komplexeren Reports und nicht zuletzt die hohen Wartungsaufwände legen eine Konsolidierung der Systeme nahe.

Evaluation ExaData
Der Evaluation wurde gegen eine ExaData V2.0 Full Rack durchgeführt. Für die Generierung eines systemnahen Lasttest wurde jede Reportanfrage auf allen Data Marts für einen Tag aufgezeichnet (Abb. 3). Der resultierende Lasttest besteht aus 21.000 Reportausführungen, mit über 1,3 Millionen SQL-Anweisungen (pro Report werden mehrere SQLs generiert) und einer maximalen Gleichzeitigkeit von 80 Sessions. Im Tagesdurchschnitt liefen 5 Sessions (aktueller Höchstwert 34) über den Tag verteilt.

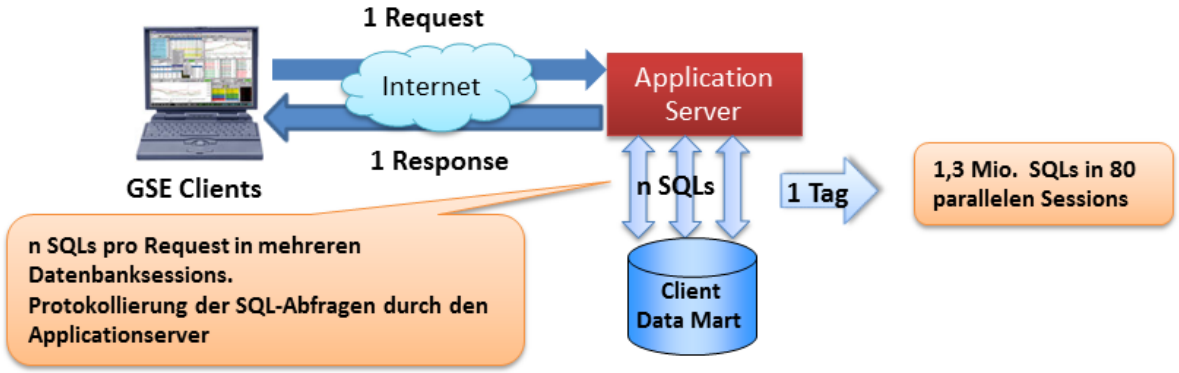

Abb. 3: Protokollierung der Kundenanfragen

Zur besseren Interpretierbarkeit der Lasttest-Messungen ist in Tabelle 1 die Summe einiger Hardwarekennzahlen aller beteiligten Data Marts gegen die Kennzahlen einer ExaData V2.0/X2-2 aufgetragen. Die an dem aufgezeichneten Lasttest beteiligten Data Mart Server haben eine in Summe fast zehnmal größere SGA und mehr als die sechsfache Menge an CPU-Cores als die Exadata V 2.0.

Merkmal	$\sum$ ***Data Mart Server***	ExaData 2.0	ExaData X 2-2
RAM /GB	3.328	576	768
SGA /GB	2.368	256	350
Cores (Compute Nodes)	416	64	96

Tabelle 1: Hardwarevergleich Data Marts versus ExaData

Die SQL-Anweisungen des Lasttests wurden auf 80/40 SQL-Files mit identischer Laufzeit verteilt. Diese SQL-Files werden zeitgleich in sqlplus gestartet. Die Sessions werden über einen Oracle Service gleichmäßig auf alle acht Knoten der ExaData verteilt. Nach 50 SQL-Anweisungen wird pro Session eine neue Verbindung initiiert, um eine gleichmäßige Auslastung der einzelnen Knoten sicherzustellen.

Als Baseline der Messungen dient die Summe der Laufzeiten aller SQLs auf den Data Marts.
Es wurden Lasttests auf der ExaData V2.0 mit unterschiedlicher Anzahl an lesenden und schreibenden Sessions auf komprimierten und unkomprimierten Daten mit und ohne `Auto DOP` gemessen (siehe Abb. 4). Dabei wurde bewusst eine hohe Anzahl an gleichzeitigen Sessions gefahren um die Skalierungsfähigkeit der ExaData zu verifizieren. Die schreibenden Sessions simulieren den Ladeprozess.

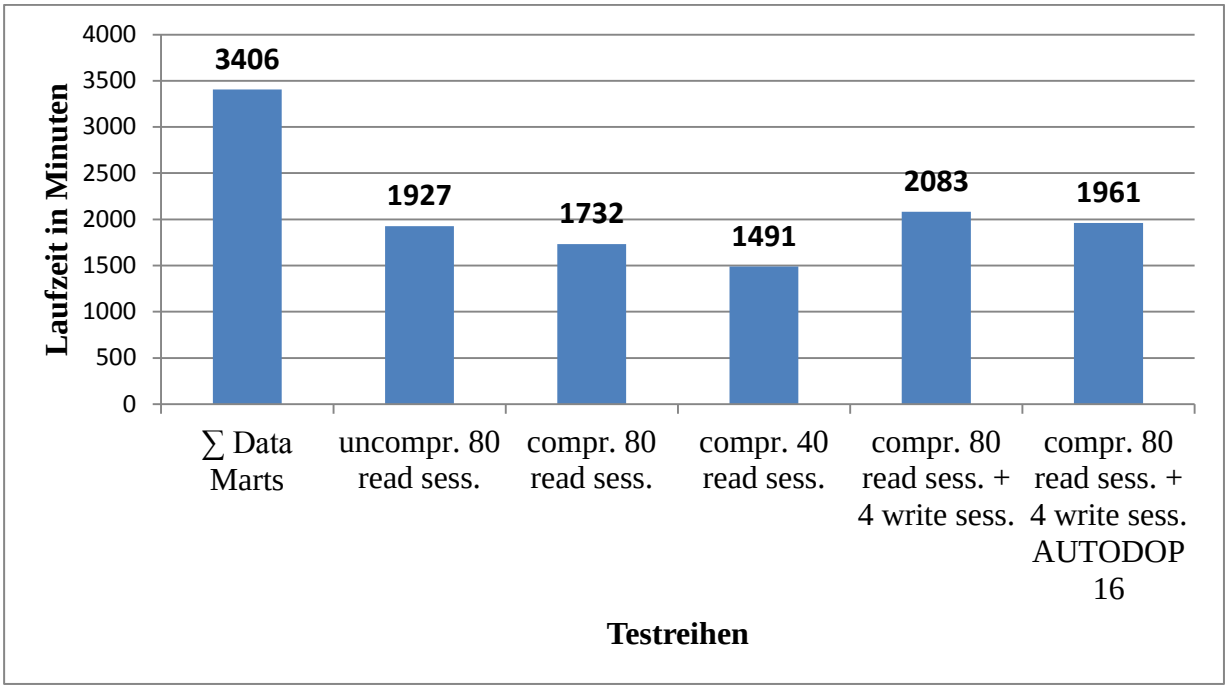

Abb. 4: SQL-Laufzeitvergleich Data´Marts ExaData V2.0

Die Summe der Laufzeiten aller SQLs war auf der ExaData selbst im ungünstigsten Testfall mit 80 lesenden und vier schreibenden Sessions deutlich besser.

Reportexecutions

Um den synthetischen Lasttest zu ergänzen, wurden an exemplarischen Reports, Ausführungszeiten auf der ExaData (ohne `Auto DOP`), und einem Data Mart verglichen (siehe Abb.5). Für Reports, bei denen die Ausführungszeit auf der ExaData größer als auf dem Data Mart waren, wurden die Ausführungen auf der ExaData mit `Auto DOP` und einem `PARALLEL_DEGREE_LIMIT` von 16 wiederholt (Abb. 5).

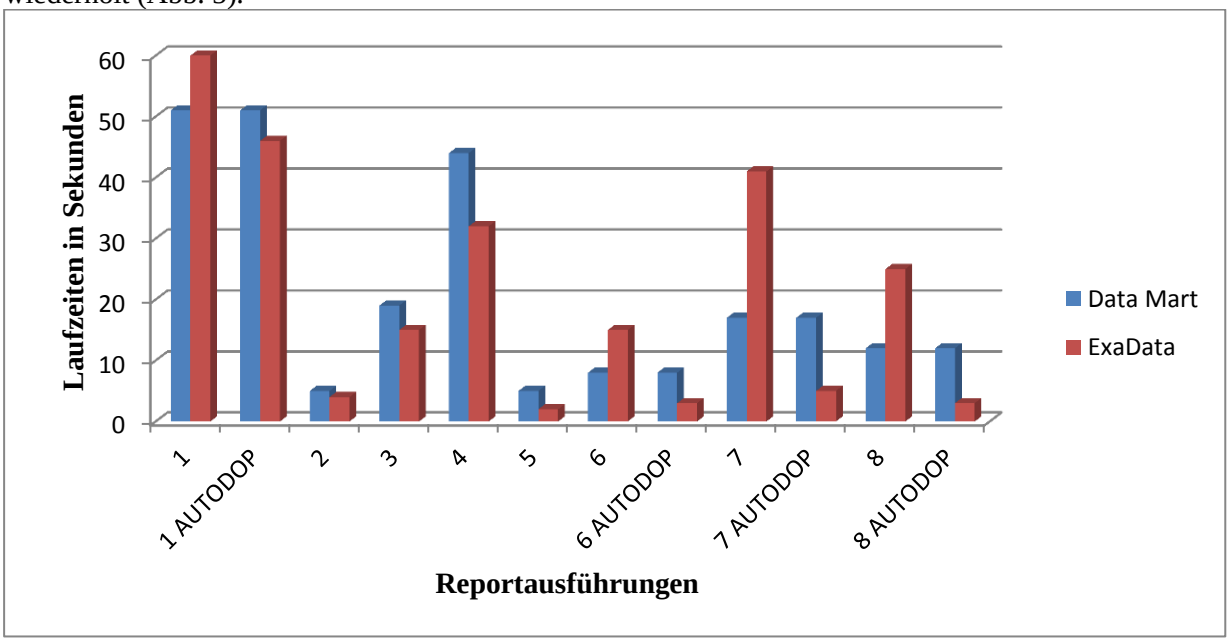

Abb. 5: Laufzeitenvergleich Reportausführungen ExaData versus Data Mart

Von den acht exemplarisch ausgewählten Reports mit Laufzeiten auf den Data Marts zwischen 5 und 60 Sekunden lief die Hälfte der Reports schneller auf der ExaData. Mit `Auto DOP` auch die restlichen Reports.

Die Ausführungszeiten der Reports spiegeln nicht das gute Ergebnis aus dem Lasttest wieder. Das hat mehrere Gründe. Zum einem die Dimensionierung der ExaData. Bei den Reporttests wurden sechs Knoten der ExaData verwendet mit jeweils einer SGA von 10 GB, der Data Mart hatte eine SGA von 70 GB. Auf der ExaData müssen mehr Daten bei der Verarbeitung gelesen werden, da die Datenmengen nicht auf einen Sektor wie beim Data Mart (z.B, Consumer Electronics) begrenzt sind. Der Zugriff auf die Daten ist über generierte SQL-Hints stark reglementiert, weshalb die Stärken der ExaData nicht voll zur Entwicklung kommen, hierauf wird später noch eingegangen. Neben der reinen Datenbankzeit auf der ExaData und dem Data Mart fließen weitere Effekte in die Gesamtlaufzeit ein. Zum einen die Metadatenanfragen zur Auflösung der Reportstruktur, die auf einer separaten Metadatenbank läuft. Zusätzlich kommt die Verarbeitungszeit innerhalb der Middleware Schicht hinzu. Des weiteren ist der I/O auf den Data Marts bei der seriellen Reportausführung erstaunlich gut, die Werte beim `sequential read` liegen hier teilweise bei nur zwei bis vier Millisekunden. Die ExaData liegt im Vergleich bei einer Millisekunde.

Fazit der Evaluation
Mit der ExaData können viele Reports gleichzeitig ohne merklichen Performanceverlust ausgeführt werden. Das Parallelisieren kann einen Report deutlich beschleunigen. Die Verwendung des `Auto DOP` vereinfacht die Parallelisierung. Die OLTP-Komprimierung ermöglicht schnelleres Lesen.
Allerdings sind bei serieller Ausführung und fehlender Gleichzeitigkeit die Reporte nicht schneller als auf den Data Marts.

Wirtschaftlichkeitsbetrachtung
Obwohl die Anschaffung einer ExaData eine erhebliche Investition darstellt, ist der Betrieb einer Data Mart Farm nicht wirklich günstiger. Die Hardware selbst schlägt bei der ExaData mit einem Listenpreis von ca. 800.000€ zu Buche. Dafür bekommt man 8 Compute Nodes, 14 Storage Nodes, fertig über Infiniband verkabelt. Die 60 Data Mart Blades, die leistungsmäßig gleichauf mit den Nodes der ExaData sind, sind für deutlich weniger zu haben. Allerdings können sie als autonome Systeme wesentlich schlechter ausgenutzt werden, was sich in einem erhöhten Lizenzbedarf für größtenteils im Leerlauf verweilende Cores niederschlägt (96 Cores ExaData vs. 576 Cores auf den Data Marts).
Am größten ist allerdings der Unterschied auf der administrativen Seite. Da jeder Data Mart einzeln aktualisiert und gewartet werden muss, ersteht ein erheblicher manueller Aufwand bei der Fehlerbeseitigung im Betrieb, bei der Bestückung und bei der Verteilung neuer Software. Eine Änderung des Schemas auf den Data Marts ist eine hochsensible, mehrere Tage dauernde Aktion für die Administratoren.

Produktivstellung der X2-2 zur Konsolidierug der Data Mart Farm
Anfang April 2011 wurde für die Konsolidierung der Data Mart Farm eine ExaData X 2-2 Full Rack geliefert. Nach Grundinstallationen von Oracle und unserem internen Dienstleister stand die Database Machine Anfang Mai 2011 mit gut drei TB komprimierten Produktivdaten (unkomprimiert knapp sechs TB) zur Verfügung.
Um die optimale Konfiguration für das Star Track System zu ermitteln wurde mit der bereits oben beschriebenen Last (80 schreibende und 4 lesende Sessions) verschieden Parametrisierungen in Bezug auf `Auto DOP` ausgeführt.

OLTP Compression

Durch den Einsatz der `OLTP Compression` konnten die Faktendaten und Hauptdimensionen von 3,7 TB auf 1,2 TB verkleinert werden.

Bei einer Simulation des Ladeprozesses, der aus `DELETE` und `INSERT` –Anweisungen besteht, wurde das Zeitverhalten von OLTP-komprimierten mit nicht komprimierten Relationen überprüft. Dabei ergaben sich längere Laufzeiten sowohl bei `DELETE`- als auch bei `INSERT`-Anweisungen bei den OLTP-komprimierten Objekten. Je nach Aufbau der Relation liegt die Performance-Degradierung bei den `DELETE` -Anweisungen zwischen 8 und 27 Prozent, bei den `INSERT` -Anweisungen zwischen 70 und 190 Prozent. Dafür werden die Daten um bis zu Faktor 4 schneller gelesen.

Da die Daten häufiger gelesen als geschrieben werden und somit die Leseperformance für das Reporting wichtiger als die Schreibperformance ist, fällt die Ladeperformanceverschlechterung nicht ins Gewicht.

OPTIMIZER_FEATURES_ENABLE

Vor den Tests mit den `Auto DOP`-Parametern wurden die Auswirkungen vom Oracle-Parameter `OPTIMZER_FEATURES_ENABLE (OFE)` auf den Lasttest ermittelt. Neben dem bisher verwendeten Wert von `10.2.0.4` wurden auch die Werte `11.2.0.1` und `11.2.0.2` getestet. Dabei zeigte sich, dass bei serieller Ausführung der SQL-Anweisungen die Laufzeitabweichungen unter sieben Prozent liegen. Bei Verwendung des `Auto DOPs` werden allerdings Laufzeitunterschiede von bis zu 37 % ersichtlich (siehe Abb. 6). Die besten Zeiten sowohl in serieller als auch paralleler Ausführung werden mit `OFE 10.2.0.4` erreicht. Die Generierung der `SQL-Hints` ist auf diese Version des Optimizers abgestimmt. Ein Test der Workload ohne Hints wurde nach eintägiger Laufzeit abgebrochen.

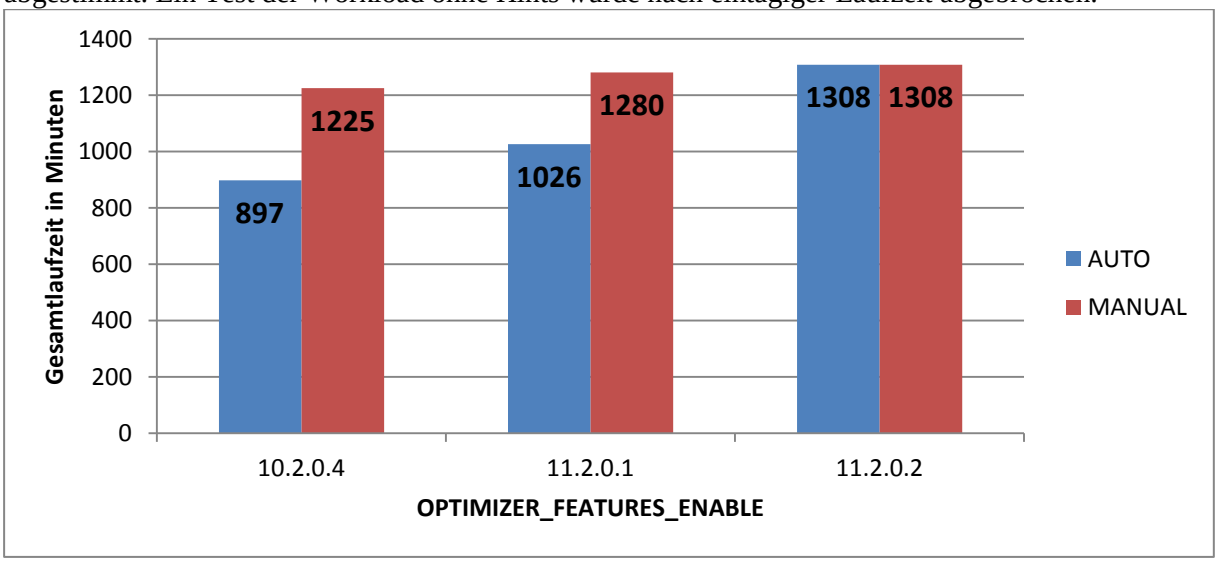

Abb. 6: Auswwirkungen optimizer_features_enable auf die Laufzeiten

Bei der Verwendung von `OFE 11.2.0.2` wurde trotz aktiviertem `Auto DOP` keine Anweisung parallelisiert, damit kommt es zu keiner Laufzeitverbesserung mit `Auto DOP`. Dies hängt mit dem Wert von `MAX_PMBS` (*„maximum megabytes per second of large I/O requests that can be sustained by a single process"*) zusammen, der beim Kalibrieren des I/O-Subsystems ermittelt und in die `sys.resource_io_calibrate$` geschrieben wird. Diesen Wert zieht der Optimizer seit der Version `11.2.0.2` für die Bestimmung des `Auto DOP` heran. Der Default-Wert lag in den Vorgängerversionen des Optimizers bei 4. Bei der Kalibrierung des Storage-Systems der ExaData wurde ein Wert von 2803 (!) bestimmt. Dieser Wert kann manuell umgesetzt werden. Der Test mit `MAX_PMBS=4` war allerdings wegen kippender Ausführungspläne ernüchternd. (siehe Abb. 7), so dass entschieden wurde, alle Berichtsanfragen unter `OFE 10.2.0.4` auszuführen.

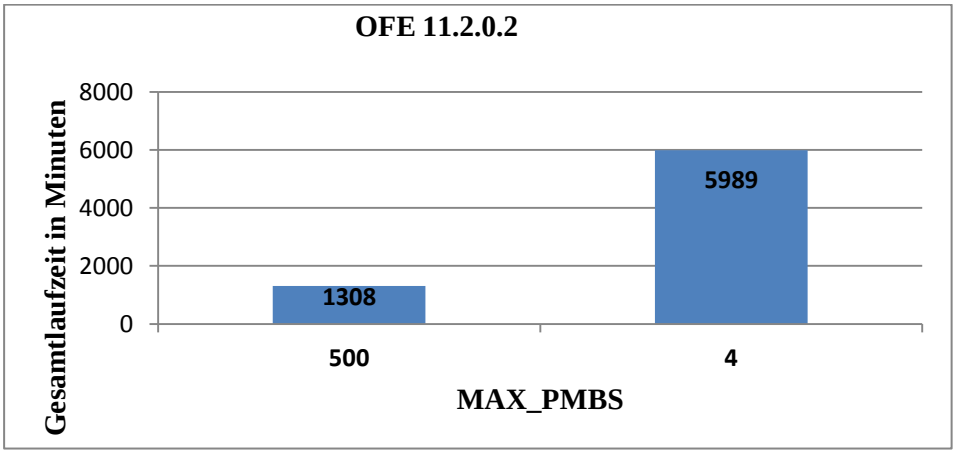

Abb. 7: Laufzeiten unter OFE 11.2.0.2 und variierenden MAX_PMBS

Auto DOP

In Abb. 8 ist die Funktionsweise des `Auto DOP` dargestellt. Neben den Parametern `PARALLEL_MAX_TIME_THRESHOLD` und `PARALLEL_DEGREE_LIMIT` wird die Auswirkung von `PARALLEL_FORCE_LOCAL` getestet, der festlegt, ob über mehrere Instanzen parallelisiert werden kann (`FALSE`) oder das SQL auf einer Instanz ausgeführt wird (`TRUE`).

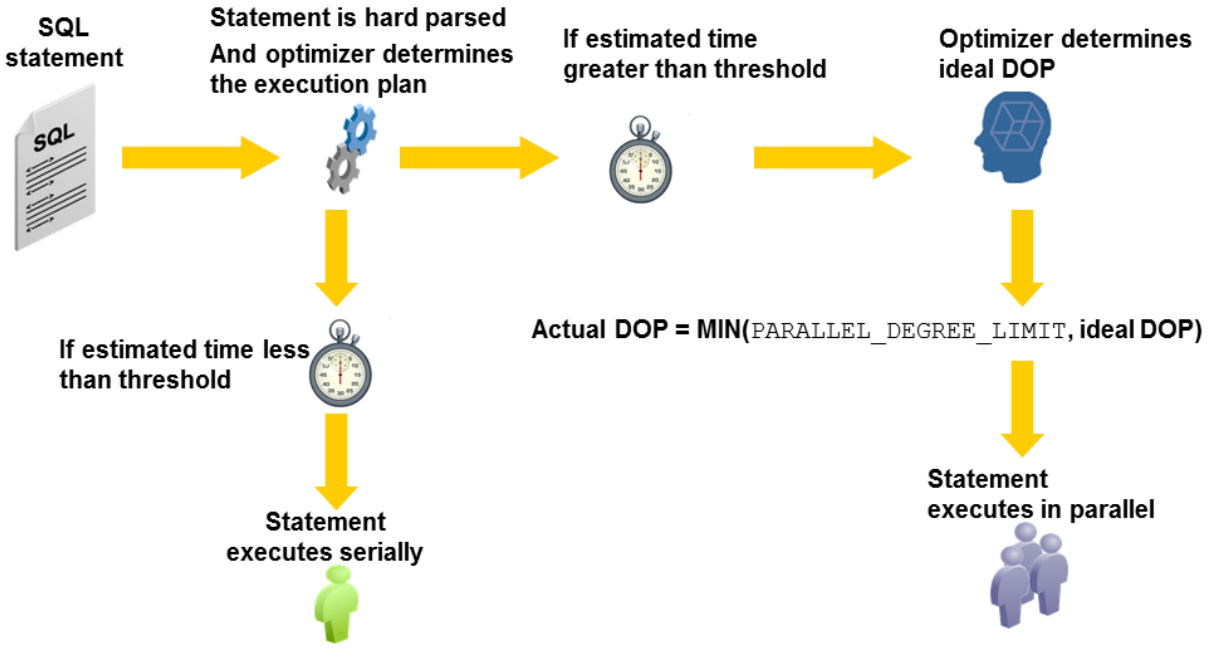

Abb. 8: Funktionsweise Auto DOP [1]

Der Parameter `PARALLEL_DEGREE_LIMIT` legt die maximale Anzahl an Prozessen fest, über die eine SQL-Anweisung parallelisiert werden kann. Es fanden Testreihen mit `PARALLEL_DEGREE_LIMIT` zwischen 2 und 64, jeweils lokal und über mehrere Instanzen verteilt, statt (Abb. 9).

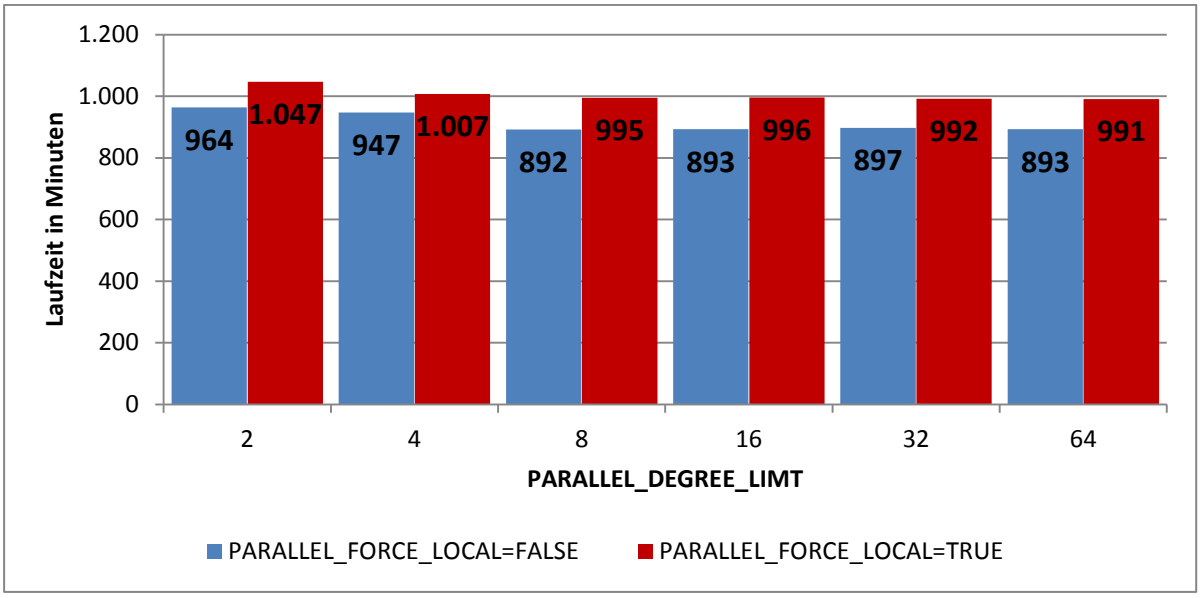

Abb. 9: Testreihe PARALLEL_DEGREE_LIMIT, PARALLEL_FORCE_LOCAL

Es zeigt sich bei der Lasttestreihe ein Optimum bei achtfacher Parallelisierung, unabhängig von lokaler oder verteilter Ausführung. Durchgängig sind die Laufzeiten bei `PARALLEL_FORCE_LOCAL=FALSE` um zehn Prozent besser als bei lokaler Parallelisierung, was durch eine bessere Lastverteilung zu erklären ist.

Um die Effektivität des `Auto DOP` zu überprüfen, wurde der Lastest soweit abgeändert, dass jedes SQL mit `PARALLEL 8` läuft und die Laufzeiten mit dem Ergebnis von oben verglichen. Es zeigt sich, dass die Verwendung des `Auto DOP` die deutlich bessere Alternative darstellt (Abb. 10).

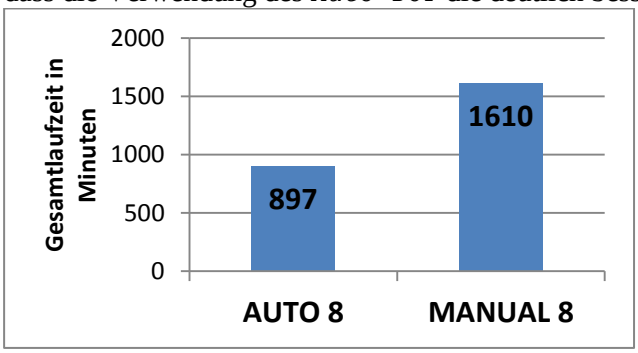

Abb. 10: Vergleich manueller DOP und Auto DOP

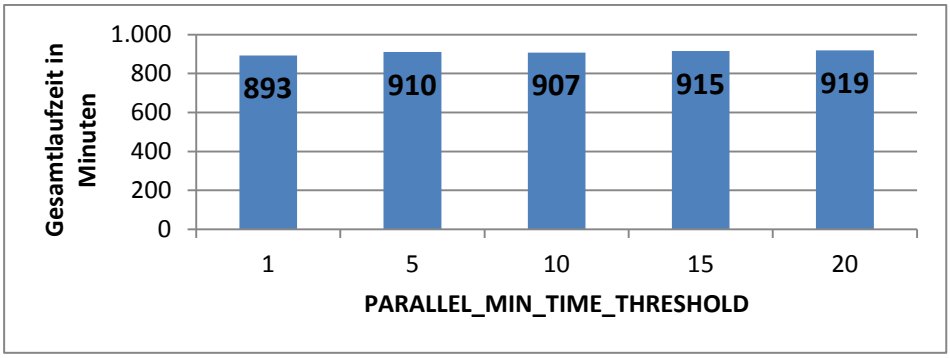

Abb. 11: Einfluss von PARALLEL_MIN_TIME_THRESHOLD auf die Laufzeit

Abb. 11 illustriert den Einfluss von `PARALLEL_MIN_TIME_THRESHOLD` auf die Gesamtlaufzeit des Lasttestes. Bei einem kleineren Wert des Parameters `PARALLEL_MIN_TIME_THRESHOLD`, werden mehr SQL-Anweisungen parallelisiert. Das führt nicht zwangsweise zu schnelleren Ausführungszeiten, da die Ausführungspläne einiger SQL-Anweisungen nicht optimal sind, bzw. der Overhead der Parallelisierung zu groß wird.

Das SQL aus Abb. 12 zeigt einen kippenden Ausführungsplan beim Parallelisieren einer SQL-Anweisung. Wird bei serieller Ausführung ein `INDEX UNIQUE SCAN` auf einem sehr selektiven Index durchgeführt, so erfolgt der Zugriff bei paralleler Ausführung über einen `INDEX RANGE SCAN` auf einen nicht selektiven Index. Damit verhundertfachen sich die `consistent gets` und damit die Ausführungszeit von 23 ms auf 3 s (siehe Abb. 13). Zur Vermeidung dieser Situationen ist es empfehlenswert `PARALLEL_MIN_TIME_THRESHOLD` nicht zu klein zu setzen. Der Default Wert des Parameters liegt bei 10 Sekunden, bei der GfK wurde ein Wert von 20 Sekunden eingestellt.

```
SELECT src.ELEMENTGROUP_ID,dst.ELEMENTGROUP_ID, src.STABLE_ELEMENTGROUP_ID
  FROM lu_ELEMENTGROUP src, #
       lu_ELEMENTGROUP dst
 WHERE src.STABLE_ELEMENTGROUP_ID = dst.STABLE_ELEMENTGROUP_ID
   AND src.GROUP_ID = dst.GROUP_ID
   AND ((src.ELEMENTGROUP_ID IN (130818,...379058)))
   AND dst.GROUP_VERSION_ID = (SELECT /*+ no_merge */ MAX(gin.GROUP_VERSION_ID)
                               FROM   lu_ELEMENTGROUP gin
                               WHERE gin.GROUP_ID = src.GROUP_ID );
```

Abb. 12: Optimizer Ausreißer bei Parallelisierung

Statistiken	seriell	00:00:00.23
1	recursive calls	
0	db block gets	
11666	consistent gets	
0	physical reads	
0	redo size	
100712	bytes sent via SQL*Net to client	
3873	bytes received via SQL*Net from client	
321	SQL*Net roundtrips to/from client	
0	sorts (memory)	
0	sorts (disk)	
4794	rows processed	

Statistiken	parallel	00:00:03.03
16	recursive calls	
0	db block gets	
918721	consistent gets	
37	physical reads	
0	redo size	
102153	bytes sent via SQL*Net to client	
3873	bytes received via SQL*Net from client	
321	SQL*Net roundtrips to/from client	
2	sorts (memory)	
0	sorts (disk)	
4794	rows processed	

Abb. 13: statistics SQL-Anweisung

Reportausführungen auf der ExaData

Während der Evaluation der ExaData wurde bereits festgestellt, dass kleine, schnelle Reports unter 5 Sekunden auf der ExaData kaum zu beschleunigen sind. Ihr außerordentliches Potenzial zeigt sie erst bei Reports, die umfangreiche Datenvolumina lesen und parallel verarbeiten. Es gibt jedoch auch Reports, bei denen die ExaData an ihre Grenzen stößt.

Grenzen der ExaData - Reportausführung über mehrere Instanzen

Eine Reportanfrage kann von der Middleware auf mehrere Datenbanksessions verteilt werden. Laufen diese Sessions auf verschiedenen Instanzen, so kommt es bei gewissen Konstellationen zu teuren Cluster Wait-Events.

In Abb. 14 laufen acht parallele Datenbanksessions einer Reportausführung auf fünf verschiedenen Instanzen. Dunkelgrün sind Cluster-, hellgrün CPU- und blau I/O-Wait-Events. Insgesamt sind 90% aller Wait-Events dieser 8 Sessions auf `CONCURRENCY` zurückzuführen. Mit über 50 Prozent schlägt das Cluster Wait Event `gc buffer busy acquire` zu Buche. Dieses indiziert, dass eine andere Session auf diesem `BUFFER` arbeitet und auf ein `GLOBAL CACHE EVENT` wartet. Der Hot Spot entsteht dadurch, dass alle acht SQLs auf dieselben Daten der Produktdimension zugreifen und nur die Faktendaten variieren. Die Ausführungszeit dieses Reports schwankt zwischen 40 und 300 Sekunden. Eine stabile Antwortzeit ist bei der Ausführung über mehrere Knoten nicht zu erhalten.

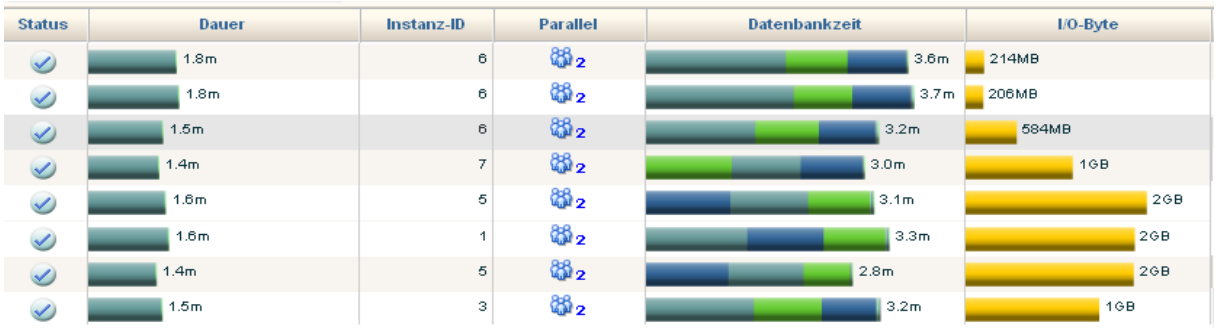

Abb. 14: Clusterwaitevents einer Reportausführung auf mehreren Instanzen

Führt man diesen Report nun auf einem dedizierten Knoten aus, so erhält man bei der ersten Ausführung ein ähnliches Bild, da erst alle benötigten Blöcke auf die dedizierte Instanz transferiert werden müssen und die parallelen Sessions sich dabei gegenseitig blockieren.
Nach wiederholter Ausführung des Reports auf einer dedizierten Instanz stabilisiert sich die Antwortzeit auf 30 Sekunden. Das Haupt-Wait-Event liegt dann auf der CPU, wegen ausgeführter HASH-Joins (siehe Abb. 15).

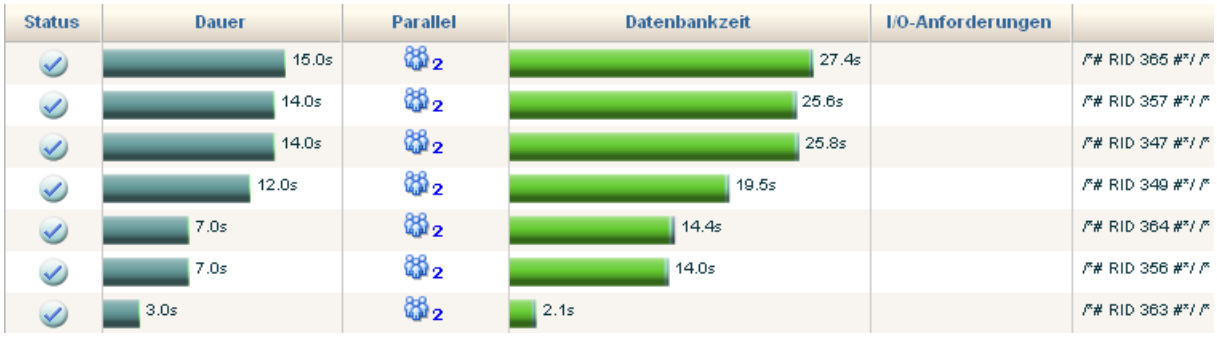

Abb. 15: Cluster Wait Events einer Reportausführung auf einer dedizierten Instanz

Dieses Beispiel legt nahe, einen Report möglichst nur auf einem Knoten auszuführen. Um `BLOCK SHIPPING` und `BLOCK CONTENTION` zu vermeiden, sollten die Daten knotenaffin abgefragt werden. Dies wird durch die Übertragung der *Sektor*-Struktur der Data Marts auf Services von Oracle erzielt. Deshalb haben wir pro *Sektor* einen Service eingerichtet, der auf eine oder zwei Instanzen beschränkt wird. Somit wird die Last auf den ExaData-Instanzen vertikal aufgeteilt. Über `PARRALEL_FORCE_LOCAL=TRUE` wird der Bericht nur auf einer Instanz ausgeführt.

Potential der ExaData

In mehreren neuen Berichtsbereichen müssen viele Warengruppen über *Sektor*-Grenzen hinweg berichtet werden. Auf den Data Marts ist das nicht möglich, da diese auf einen *Sektor* beschränkt sind.

Auf dem alten Zentralsystem sind diese Berichte oft nach mehreren Stunden wegen `SNAPSHOT TOO OLD` abgebrochen und wegen Ressourcenmangel konnte die Reportausführung nicht parallel laufen.
Auf der ExaData läuft ein Bericht dieser Art in serieller Ausführung in 25 Minuten (1500 Sekunden). Führt man diesen Report mit Auto DOP aus, so ergeben sich Laufzeiten je nach Parallelisierungsgrad zwischen 36 und 340 Sekunden (siehe Abb. 16).

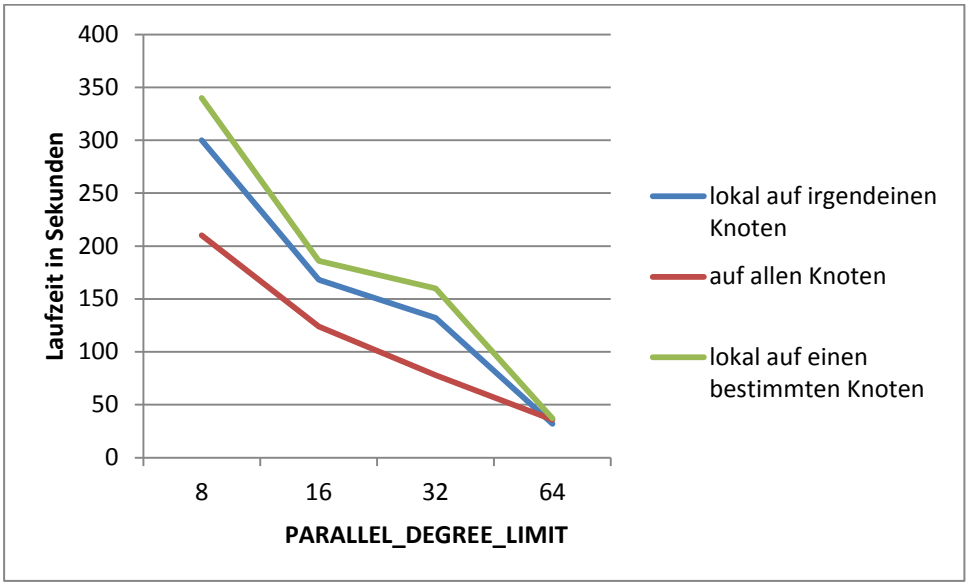

Abb. 16: Laufzeiten TSR-Report in Abhängigkeit Parallelitätsgrad und Knotenaffinitäten

Die Ausführungszeit ist minimal bei einer Parallelisierung über 64 Sessions. Entscheidend ist die Änderung des Ausführungsplans bei höherer Parallelität. Ab 64-facher Parallelisierung erfolgt der Datenzugriff nicht mehr über Bitmap-Indizes, sondern über `SMART Scan`.
Dass der Report so lange auf den Bitmap-Indizes arbeitet, liegt an den Einstellungen des Systems. So ist z.B. `OPTIMIZER_INDEX_COST_ADJ` auf 5 eingestellt. Zusätzlich werden bei den SQL-Anweisungen extrem viele `INDEX-Hints` verwendet. Im Zuge der neuen Anwendungsfälle und Einführung der ExaData wurde die SQL-Generierung soweit überarbeitet, dass ab einer gewissen Datenmenge die `INDEX-Hint` Generierung unterdrückt werden kann. Momentan gibt es Bestrebungen ein Regelwerk zu etablieren, bei dem `INDEX-` bzw. `NO_INDEX-Hints` in Abhängigkeit der Länge der `IN`-Listen in den Filterbedingungen generiert werden. Zusätzlich wird die Generierung von `FULL Hints` erwogen, um `SMART Scans` zu erzwingen.

Fazit

`Out of the box` löst die ExaData für das interaktive Kundenreporting der GfK nicht alle Probleme. Nach umfangreichen Analysen und Tests wurden diverse Anpassungen, sei es nun Parametrisierungen, Servicedefinitionen, Ressourcenmanagement aber auch Änderungen an den SQL-Anweisungen vorgenommen, um das volle Potential der ExaData zu nutzen. Einige Anpassungen müssen noch evaluiert werden, wie z.B. die Verwendung von FULL-Hints um `SMART Scans` zu provozieren.

Momentan wurden 90 Prozent aller Kunden erfolgreich auf die ExaData umgezogen. Die bisherige Auslastung in Stoßzeiten war dabei selten über 50 Prozent der CPU-Kerne. Bis Ende Oktober sollen alle Kunden umgestellt sein. Zur DOAG 2011 wird es eine verlässliche Aussage über den Erfolg der Konsolidierung geben.

Literatur

[1] „Workload Management for an Operational Data WarehouseOracle Database 11.2.0.2“, Jean-Pierre Dijcks DOAG 2010

[2] „In-Memory DWH Reporting mit Oracle”, Dr. Jens Albrecht, Marc Fiedler, Oliver Scheibe DOAG 2009

Kontaktadresse:

Oliver Scheibe, Jens Albrecht, Marc Fiedler
Nordwestring 101
D-90419 Nürnberg

Telefon: +49(0)911 395 4039
Fax: +49(0)911 395 54039
E-Mail { oliver.scheibe, jens.albrecht, marc.fiedler}@gfk.com
Internet: www.gfkrt.com